Documentación y consumo API

Lance Kenyer Kaleth Torres Bonilla

Tomas Camilo Saravia Martinez

Julian Santiago Sanchez Muñoz

Sebastian Andres Alfonso Latorre

Instructor: Javier Yara

Análisis y desarrollo de software

CEET

Bogotá, 2023

Contenido

[Introducción 3](#_Toc152968738)

[Objetivo 3](#_Toc152968739)

[Desarrollo 3](#_Toc152968740)

[Métodos: 4](#_Toc152968741)

[Conclusión 10](#_Toc152968742)

# Introducción

En este proyecto se utiliza una Interfaz de programación de aplicaciones (API) para realizar un proceso de conversión y manejo de datos desde un formato JSON apoyados de una extensión usada desde VISUAL STUDIO CODE llamada THUNDER CLIENT, que migra estos hacia una base de datos previamente creada en MYSQL.

Se entiende que THUNDER CLIENT es una extensión que ofrece una interfaz similar a POSTMAN, que es conocida como una plataforma API, permitiendo crear un puente para migrar datos con distintos métodos los cuales serán aclarados a continuación.

# Objetivo

Optimizar tiempo a la hora de manejar la información dentro de la base de datos, y crear otra alternativa rápida y eficaz para evitar el ingreso engorroso de datos en la interfaz de MYSQL.

# Desarrollo

Antes de iniciar con el proceso se creó una base de datos la cual cuenta con 19 tablas diferentes que están relacionadas una con la otra, lo cual es parte esencial para que la base de datos tenga una comunicación y una relación entre cada una de ellas.

Una creada la base de datos para implementar la ejecución en esta parte del proyecto se inició con la previa descarga de VISUAL STUDIO CODE, una vez ejecutada se instala dentro de esta, la extensión THUNDER CLIENT.

Realizado este proceso nos dirigimos a colocar el correspondiente enlace que contiene la tabla de la base de datos directamente desde MYSQL. Una vez el enlace haya quedado ingresado, crearemos un index en VISUAL STUDIO CODE que se llamara como la tabla a la que se ha sido asignada seguido de (.php).

Una vez creado dicho archivo se empezará a crear el código basado en JSON en donde empezaremos a manejar la base de datos.

En este caso se usaron cuatro métodos los cuales fueron GET, POST, PUT y DELETE los cuales tienen diferentes usos específicos.

# Métodos:

1. GET: Este método cumple la función de traer los datos, leer y mostrarlos.

A continuación, se mostrarán 3 tablas de ejemplo sobre las cuales este método fue usado en el proyecto.

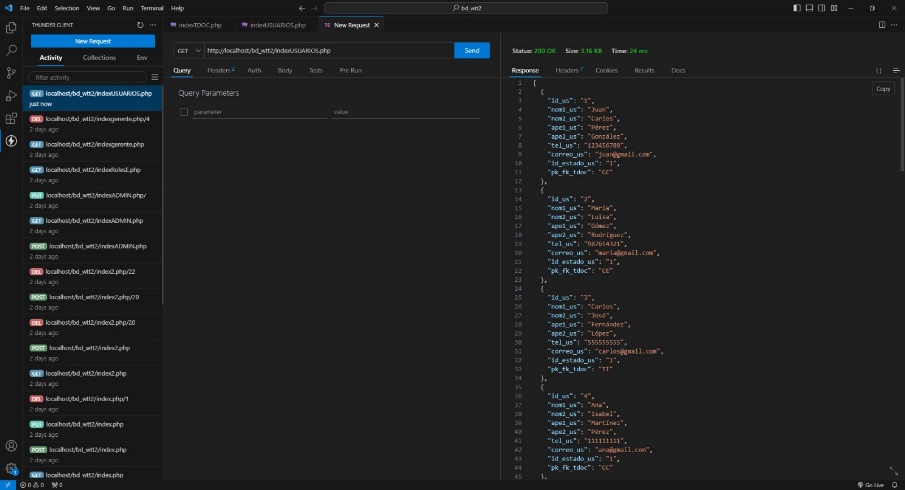


Figura 1

*Nota: Tabla usuarios*

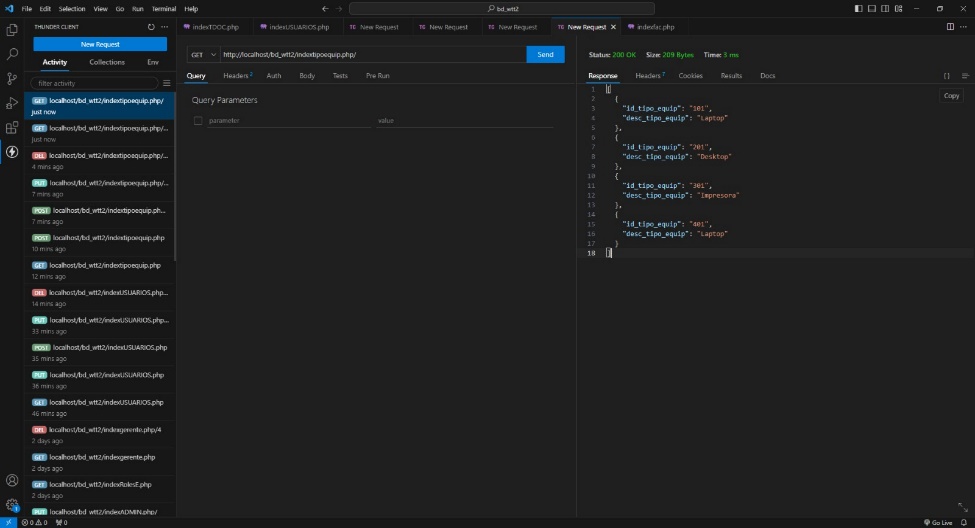


Figura 2

*Nota: Tabla tipo\_equipo*

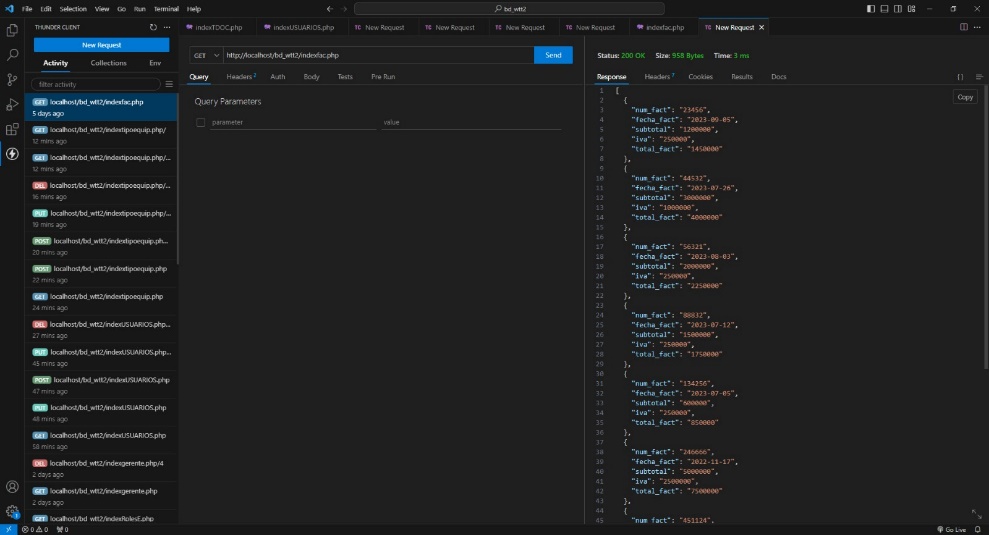
**

Figura 3

*Nota: Tabla factura*

1. POST: Con este método se implementa la función de crear nuevos datos dentro de la tabla que se esté usando.

Para este método se mostrarán tres ejemplos usando las mismas tablas usadas anteriormente, evidenciando el uso de este.

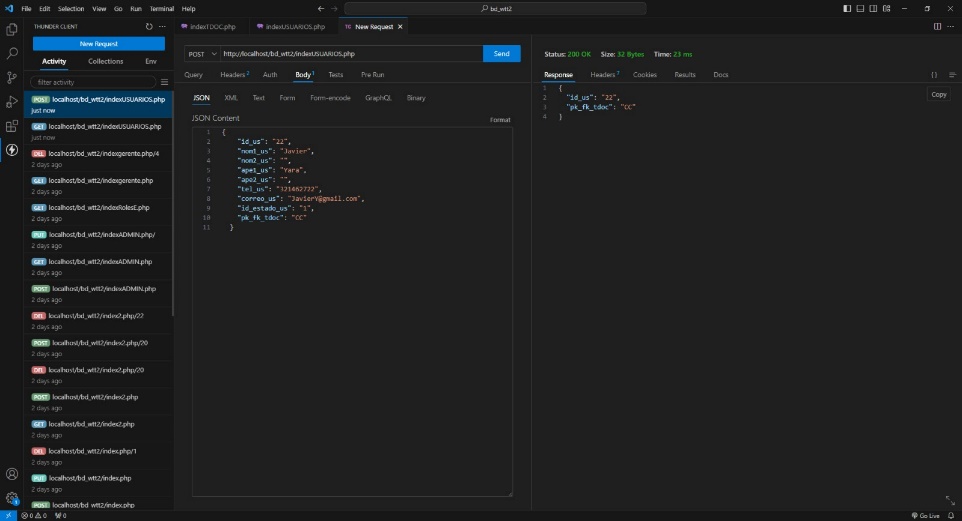


Figura 4

*Nota: Tabla usuarios*

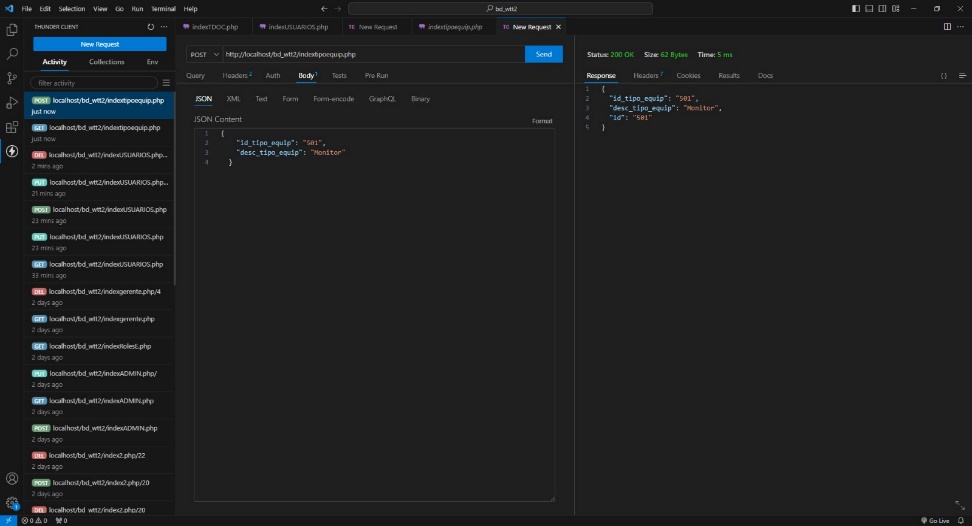
**

Figura 5

*Nota: Tabla tipo\_equipo*

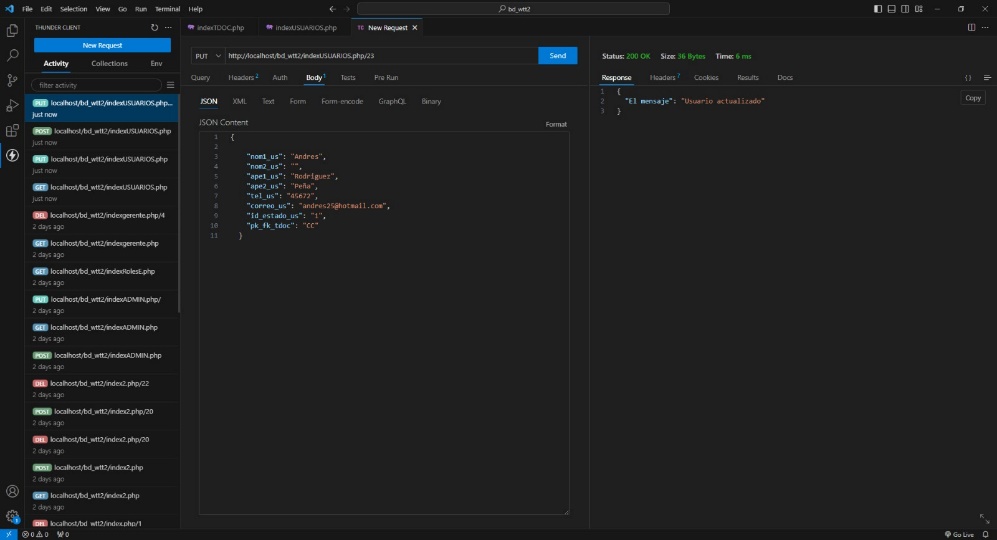
**

Figura 6

*Nota: Tabla factura*

1. PUT: Este método permite actualizar o cambiar los datos de la base de datos.

Se presentarán tres ejemplos utilizando las tablas previamente mencionadas para ilustrar la aplicación de este método.

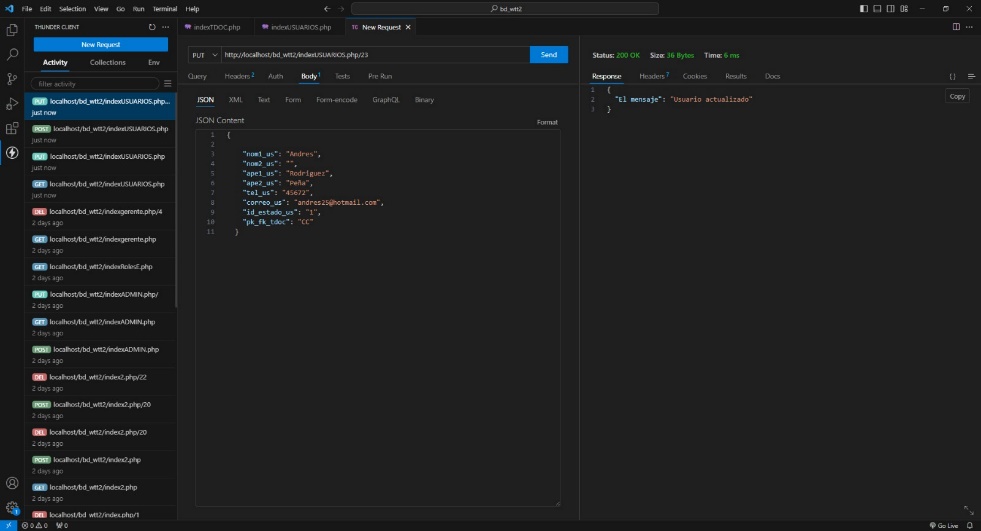


Figura 7

*Nota: Tabla usuarios*

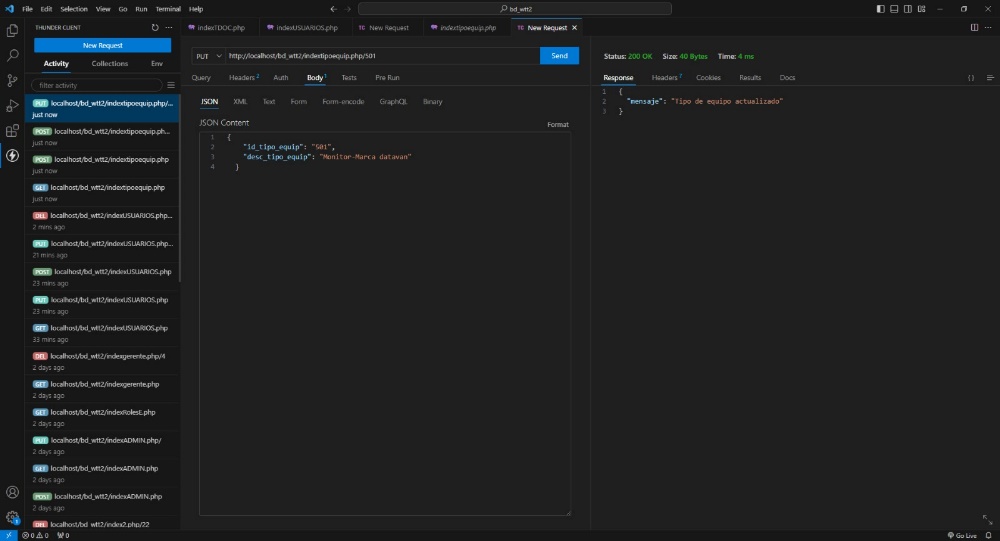
**

Figura 8

*Nota: Tabla tipo\_equipo*

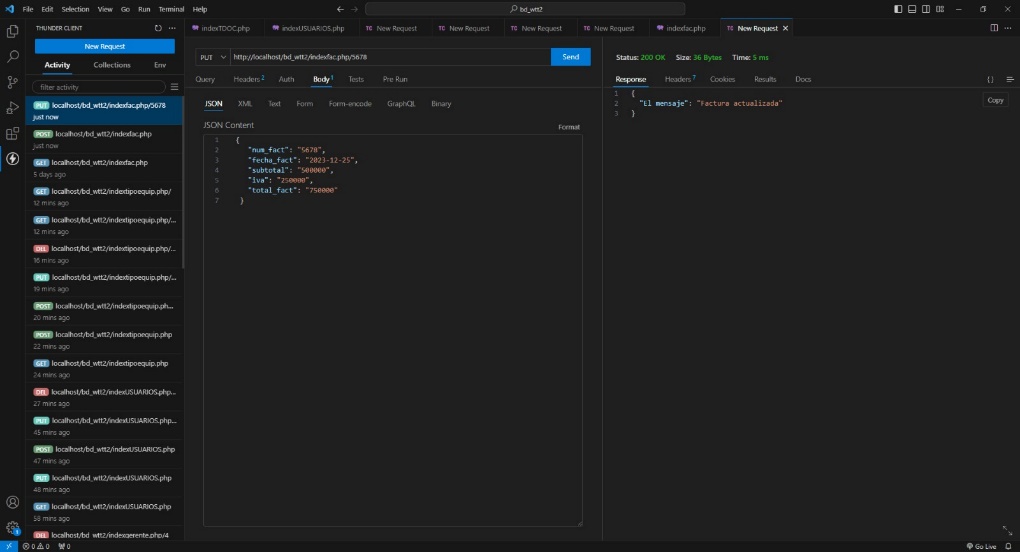
**

Figura 9

*Nota: Tabla factura*

1. DELETE: Con este método se eliminan datos de la tabla de la base de datos que se esté usando.

Se exhibirán tres instancias que emplean las tablas ya mencionadas con el propósito de demostrar la utilización de este método.

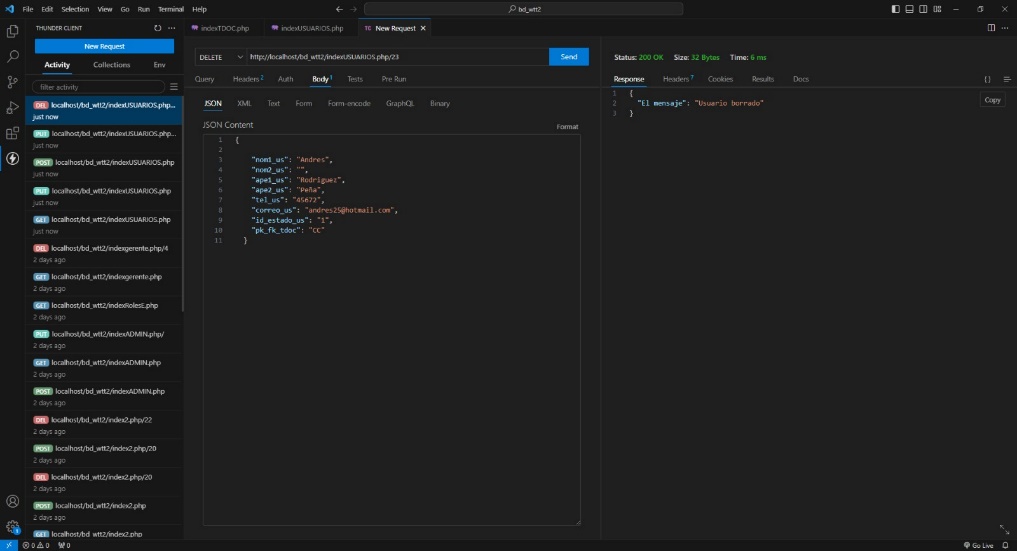


Figura 10

*Nota: Tabla usuarios*

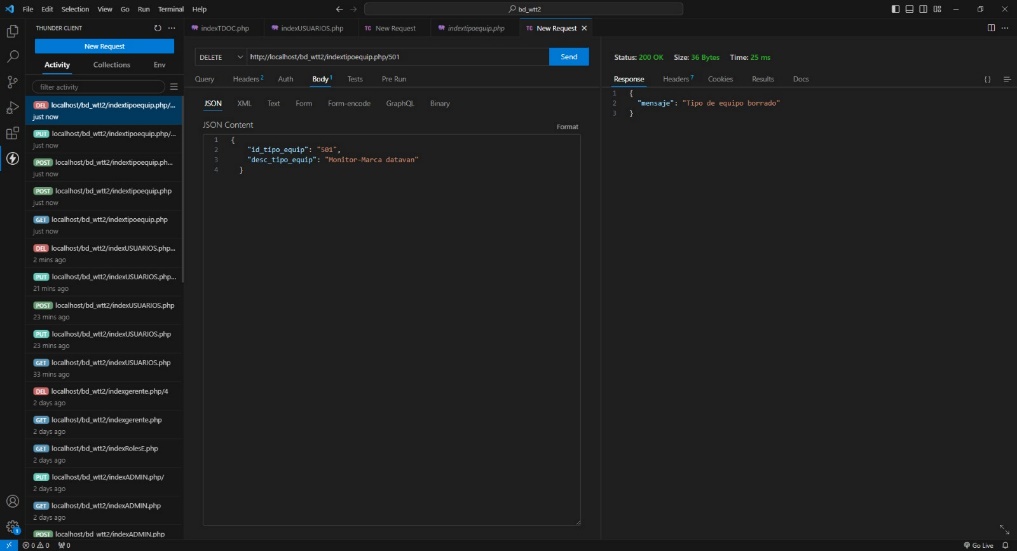


Figura 11

*Nota: Tabla tipo\_equipo*

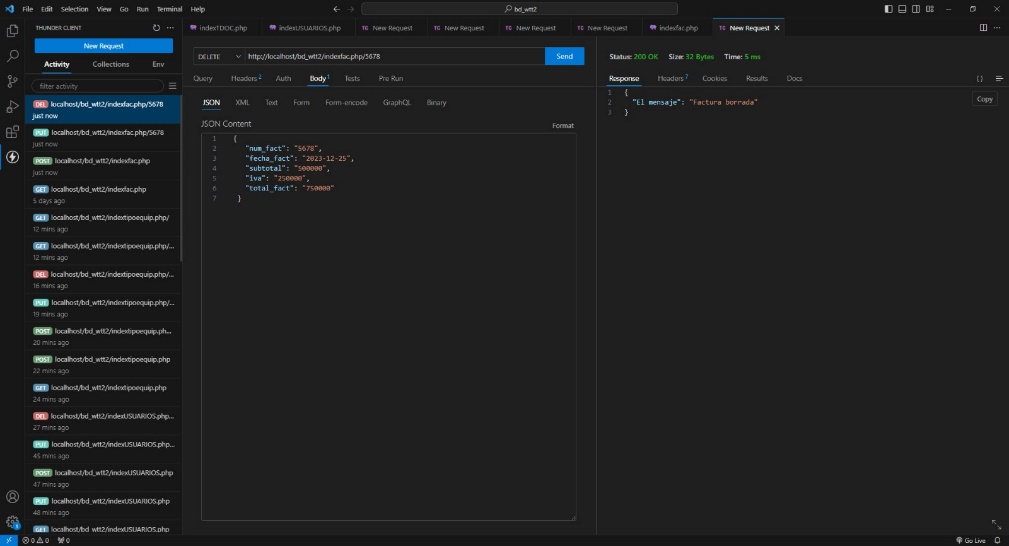
**

Figura 12

*Nota: Tabla factura*

# Conclusión

En conclusión, este proyecto ha logrado implementar de manera eficaz la Interfaz de programación de aplicaciones (API) a través de la extensión THUNDER CLIENT en VISUAL STUDIO CODE, facilitando el proceso de conversión y manejo de datos desde un formato JSON hacia una base de datos MYSQL previamente creada. La creación de 19 tablas interrelacionadas en la base de datos ha sido fundamental para garantizar una comunicación fluida entre ellas.

El objetivo principal de optimizar el tiempo en el manejo de la información se ha alcanzado con éxito, proporcionando una alternativa rápida y eficiente para evitar el ingreso tedioso de datos a través de la interfaz de MYSQL. La ejecución de los métodos GET, POST, PUT, y DELETE ha sido claramente ejemplificada mediante la manipulación de diversas tablas, demostrando su aplicación específica en la lectura, creación, actualización y eliminación de datos.

En resumen, este proyecto no solo ha mejorado la eficiencia en la gestión de datos, sino que también ha establecido un marco sólido para futuras implementaciones y desarrollos en el ámbito de la programación y manipulación de bases de datos.